Ankit chaudhary

IT_Tools FY IT - II

✱ PEP 8 in python

→ The Pep is an abbreviation form of Python Enterprise Proposal.

→ Writing code with proper logic is a key factor of programming, but many other important factors can affect the code's quality.

→ Adaptive a nice coding style makes the code more readable.

→ PEP 8 is a document that provides various guidelines to write the readable in Python.

✱ Why PEP 8 is Important?

→ PEP 8 enhances the readability of the Python code, ~~b/w~~

→ We must have an idea of why we ~~wrote~~ wrote the particular line in the code.

→ The code should reflect the meaning of each line.

Ankit chaudhary
FYIT_11

★ Naming convention

→ When we write the code, we need to assign name to many things such as variables, Functions, classes, packages.

→ When we look back to the file after sometimes we can easily recall what a certain variable, function, or class represents.

example

→ single lowercase letter
a = 10

→ single upper case letter
A = 10

→ Lower_case_with_under scores
number_of_apple = 5

★ Name style :-

→ Function :- we should use the lowercase word or separates word by the underscore.

ex myfunction, my_firstfunction

→ variable :- We should use a lowercase or separate word to enhance the readability.

ex a, var, variable_name.

* class :- The first letter of class Name should be capitalized ; use camel case. Do not separate words with the underscore.

ex MyClass , Form , Model

* Method :- We should use a lowercase , letter, words, or separate words to enhance readability.

ex class_method

* Constant :- We should use short, uppercase , letter , words, or separate word to enhance the readability

ex MYCONSTANT, CONSTANT

* Module :- we should use a lowercase letter, words , or separate word to enhance the readbility.

* Indentation :-

→ The indentation is used to define the code block in Python.

→ The indentations are the important part of

The python programming language and it determines the level of lines of code.

example

```
x = 5
if x == 5:
        print ('x and 5 are same')
```

→ This indentation defines the code block and tells us what statements execute when a functions is called or condition trigger

☆ Tabs Vs Space

☆ use The Tabs to provide the consecutive spaces to indicate the Indentation, but whitespaces are the most preferable.

→ Python 2 allows the mixing of tabs and spaces but we will get an error in python3.

☆ Indentation following Line Break.

→ It is essential to use Indentation when using line countinuations to keep the line to fewer than 79 characters.

→ It provides the flexibility to determining between two lines of code and a single line of code that extends two lines.

* Use docstring :-

→ Python provides the two types of document strings or doc string - single line and multiple lines.

→ We use the triple quotes to defin a single line or multiline quotes

ex

```
print (" single line print Statment")
print ("""" multiple line
          print
          statement""")
```

→ Python allows us to break line before or after a binary operator, as long as the convention is consistent locally.

* Importing module

import pygame ~~&~~ right
Import os

* import sys, os → wrong.

→ The import statement should be written at the top of the file or just after ~~a~~ any module comment

→ Absolute imports are the recommended because they are more readable and tend to be better behaved.

☆ Blank lines

→ Blank lines can be improved the readability of python code

→ If many lines of code bunched together the code will become harder to read.

ex
```
class First_class:
    pass

class SecondClass:
    pass.

def main_Function():
    return None
```

☆ Single blank line inside classes:

→ The functions that we define in the class is related to one another.

```
class FirstClass:
    def Method_one(self):
        return None
    def second_two(self):
        return None.
```

Ankit Chaudhary
FYIT-11

classmate Date Page

* use blank lines inside the functions:-

→ Sometimes, we need to write a complicated functions has consists of several steps before the return statement

```
def cal_variance(n_list):
    list_sum = 0
    for n in n_list:
        list_sum = list_sum + n
    mean = list_sum / len(n_list)

    square_sum = 0
    for n in n_list:
        square_sum = square_sum + n**2
    mean_squares = squarare_sum / len(n_list)

    return mean_squares - mean**2
```

* Put the closing Braces:-

→ We can break line inside parentheses, brackets using the line continuations.

→ PEP 8 allows us to use closing braces in implies line continuations.

ex

```
list_num = [
    5, 4, 1
    4, 6, 3,
    7, 8, 9
]
```

* Comments :-

→ Comments are the integral part of the any programming language.

→ These are the best way to explain the code.

→ When we documented our code with the proper comments anyone can able to understand to code.

→ Indent block comment should be at the same level

→ start each line with the # followed by a single space.

→ separate line using the single #.

```
for i in range (0,5):
    # loop will iterate over i five times and
      print out the value of i
    # New line character

    print (i, '\n')
```

* Inline comments.

→ Inline comment are used to explain the single statement in a piece of code.

→ We can quickly get the idea of why we wrote that particular line of code.

→ Start comment with the # and single space.

→ Use inline comments carefully.

→ We should separate the inline comments on the same line as the statement they refer.

```
a = 10  # The a is variable that holds
          integer value.
```

→ Avoid Unnecessary Adding whitespaces.

→ Use of whitespaces can make the code much harder to read.

→ Too much white space can mate code overly sparse and difficult to understand.

→ We should avoid adding whiteSpaces at the end of a line. This is known as trailing whitespaces.

```
# Recomended
list1 = [1, 2, 3]

# Recomended -> Not
list 2 = [1, 2, 3,]
```

```
x = 5
y = 6

# Recommended
  print (x,y)
# Not recommended
  print(x , y)
```